МИНОБРНАУКИ РОССИИ

ФГБОУ ВО Череповецкий государственный университет

Институт информационных технологий

Кафедра: Математическое и программное

обеспечение ЭВМ

Дисциплина: Объектно-ориентированное

программирование

ЛАБОРАТОРНАЯ РАБОТА №10

**ПРОГРАММИРОВАНИЕ ОБРАБОТКИ**

**ИСКЛЮЧИТЕЛЬНЫХ СИТУАЦИЙ**

Выполнил:

студент гр.1ИСб-00-21оп

Лебедева А. В.

Проверил:

Шаханов Н. И.

Череповец, 2020 г

Цель работы: изучить синтаксис и семантику определения и вызова исключений, синтаксис обработчика и спецификации исключений; приобрести практические навыки запуска исключений; изучить особенности применения стандартных библиотечных исключений.

Цель работы: изучить синтаксис и семантику определения и вызова исключений, синтаксис обработчика и спецификации исключений; приобрести практические навыки запуска исключений; изучить особенности применения стандартных библиотечных исключений.

**Задание**

Используя модифицированный АТД, обработайте все возможные исключительные ситуации

Текст программы:

#include <iostream>

#include <string>

using namespace std;

class form;

class Optica;

class form

{

protected:

string tip; //тип

string appointment; // назначение

public:

form(); // констр. по умолчанию

form(string t, string a); // конструктор

form(const form&);

//~form(); // деструктор

void printF() const; //селектор

};

class Optica : public form

{

private:

string linz;

double power;

double veight; // вес

public:

//Optica(); // констр. по умолчанию

Optica(const Optica&); // копирующий конструктор

Optica(string t, string a, string l, int p, float v); // конструктор с параметрами

// ~Optica(); // деструктор

void print() const;// селектор

Optica& operator = (const Optica& obj); // присваивание

bool& operator == (Optica& obj); // сравнение

};

form::form()

{

tip = "tip";

appointment = "appointment";

}

form::form(string t, string a)

{

if (t == "") throw 1;

if (a == "") throw 2;

tip = t;

appointment = a;

}

void form::printF() const

{

cout << "Тип: " << tip << endl;

cout << "Назначение: " << appointment << endl;

}

form::form(const form& obj)

{

tip = "";

appointment = "";

tip = obj.tip;

appointment = obj.appointment;

}

Optica::Optica(string t, string a, string l, int p, float v) : form(t, a)

{

if (l == "") throw 3;

if (p <= 0 || v <= 0) throw 4;

linz = l;

power = p;

veight = v;

}

Optica::Optica(const Optica& obj)

{

linz = obj.linz;

power = obj.power;

veight = obj.veight;

}

void Optica::print() const

{

cout << "Линза: " << linz << endl;

cout << "Мощност: " << power << endl;

cout << "Вес: " << veight << endl << endl;

}

Optica& Optica :: operator=(const Optica& obj)

{

if (this != &obj)

{

tip = ""; appointment = "";

linz = "";

tip = obj.tip;

appointment = obj.appointment;

linz = obj.linz;

power = obj.power;

veight = obj.veight;

}

return (\*this);

}

bool& Optica :: operator== (Optica& obj)

{

bool flag = true;

int a = 0;

if (tip == obj.tip) a++;

if (appointment == obj.appointment) a++;

if (linz == obj.linz) a++;

if (power == obj.power) a++;

if (veight == obj.veight) a++;

if (a == 5) flag = true; else flag = false;

return flag;

}

int main()

{

setlocale(LC\_ALL, "Rus");

try

{

Optica a("", "Ближнего действия", "Выпуклая", -13, 0.2);

Optica b("Фотоаппарат", "Дальнего действия", "", 13, 0.2);

a.printF();

a.print();

b.print();

}

catch (int i)

{

if (i == 1) cout << "Не указан тип оптического устройства." << endl;

if (i == 2) cout << "Не указано назначение оптического устройства." << endl;

if (i == 3) cout << "Не указан тип линзы оптического устройства." << endl;

if (i == 4) cout << "Некорректное значение." << endl;

}

system("pause");

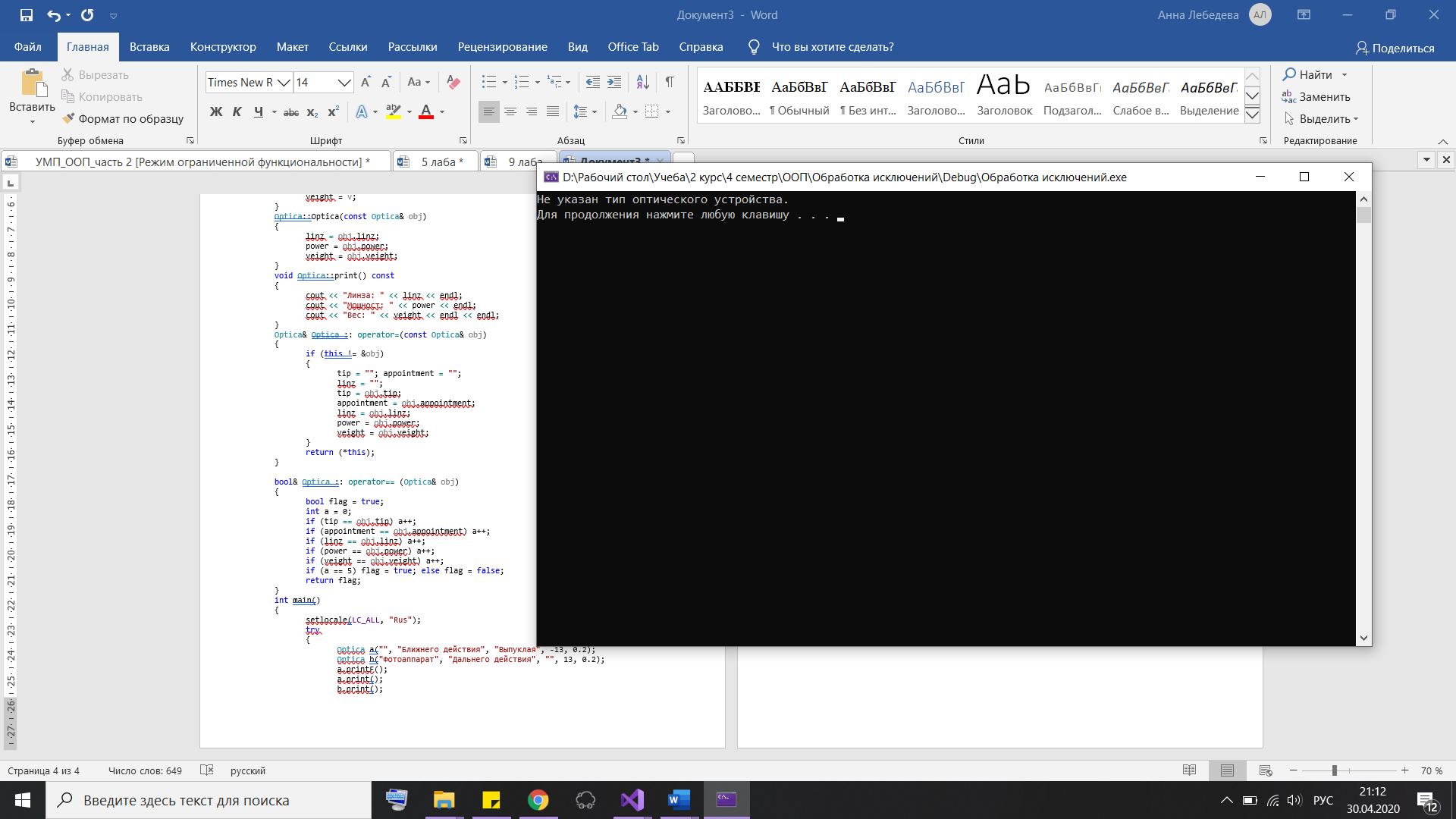
return 0;

}

Вывод:

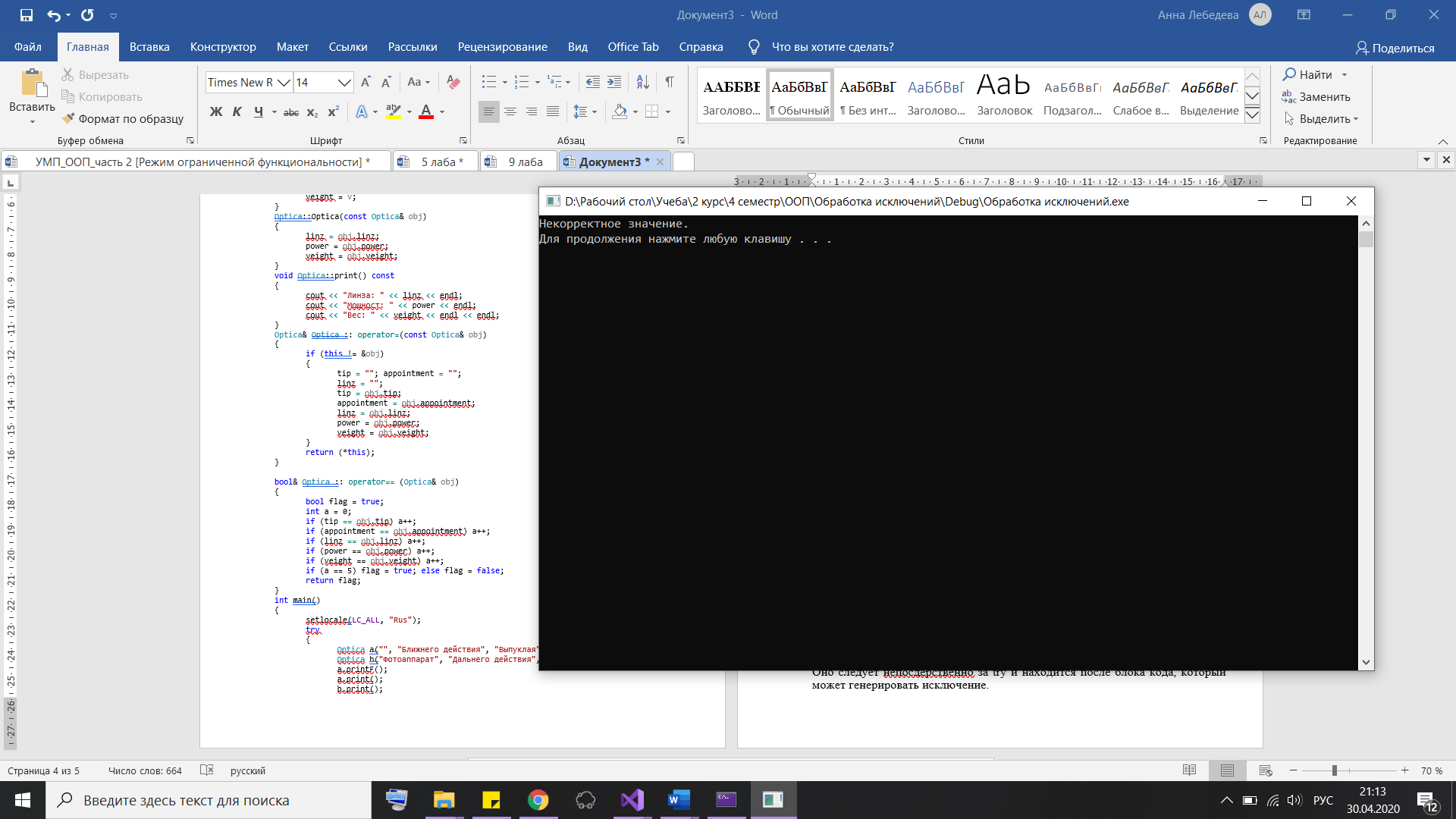
1. Optica a("", "Ближнего действия", "Выпуклая", -13, 0.2);

Optica b("Фотоаппарат", "Дальнего действия", "", 13, 0.2);



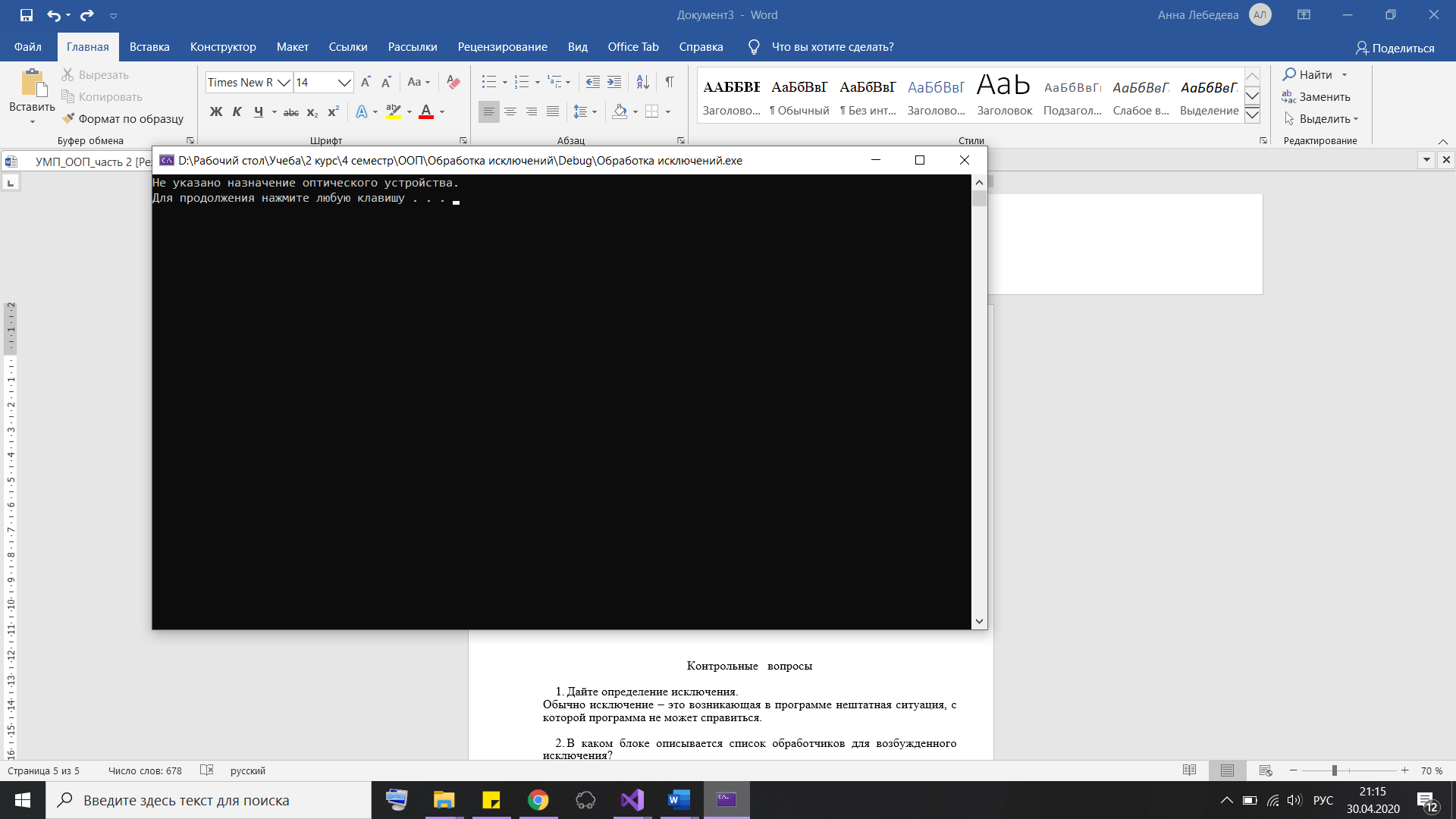
1. Optica a("Лупа", "Ближнего действия", "Выпуклая", -13, 0.2);

Optica b("Фотоаппарат", "Дальнего действия", "", 13, 0.2);



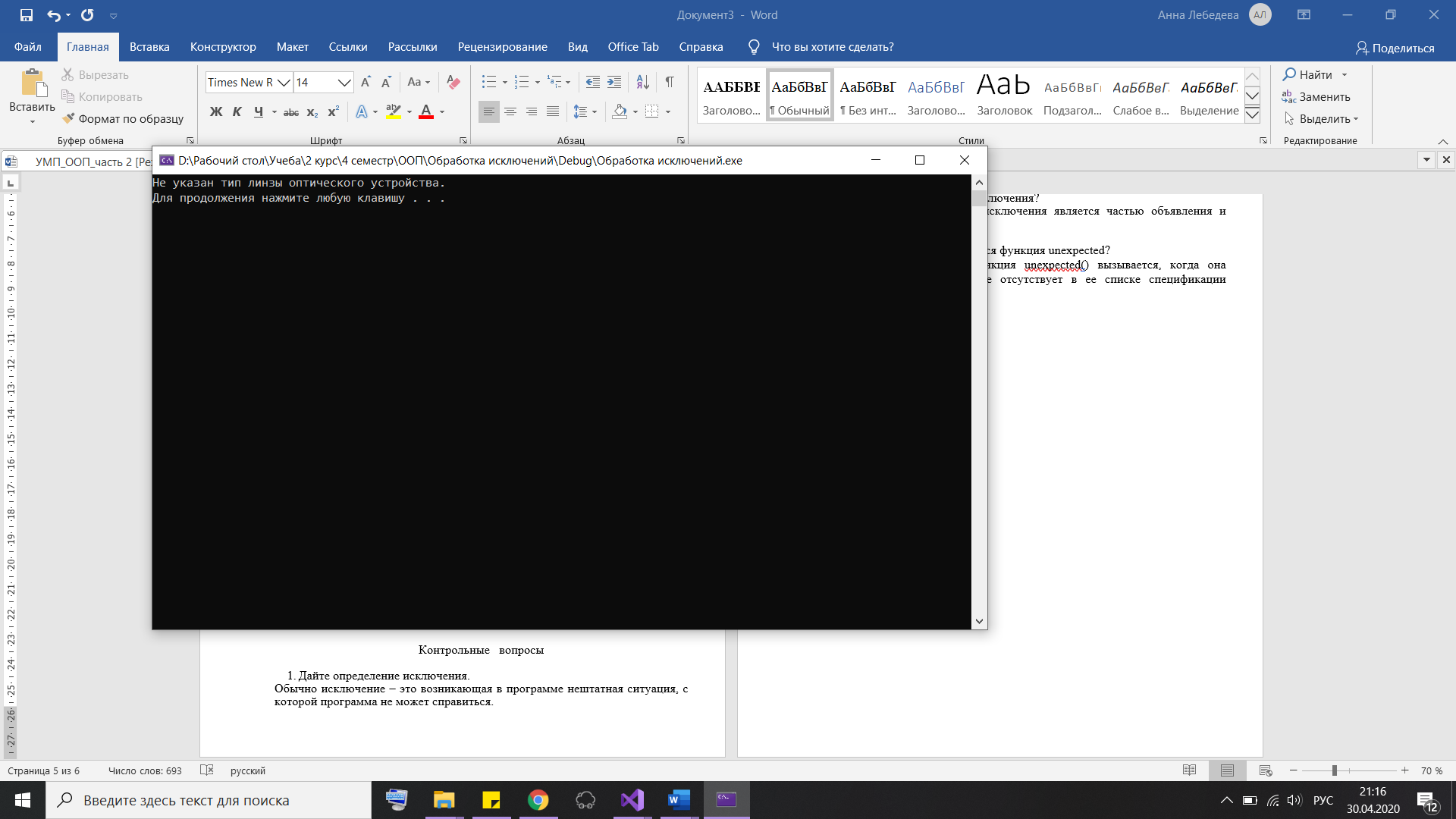
1. Optica a("Лупа", "Ближнего действия", "Выпуклая", 13, 0.2);

Optica b("Фотоаппарат", "", "", 13, 0);



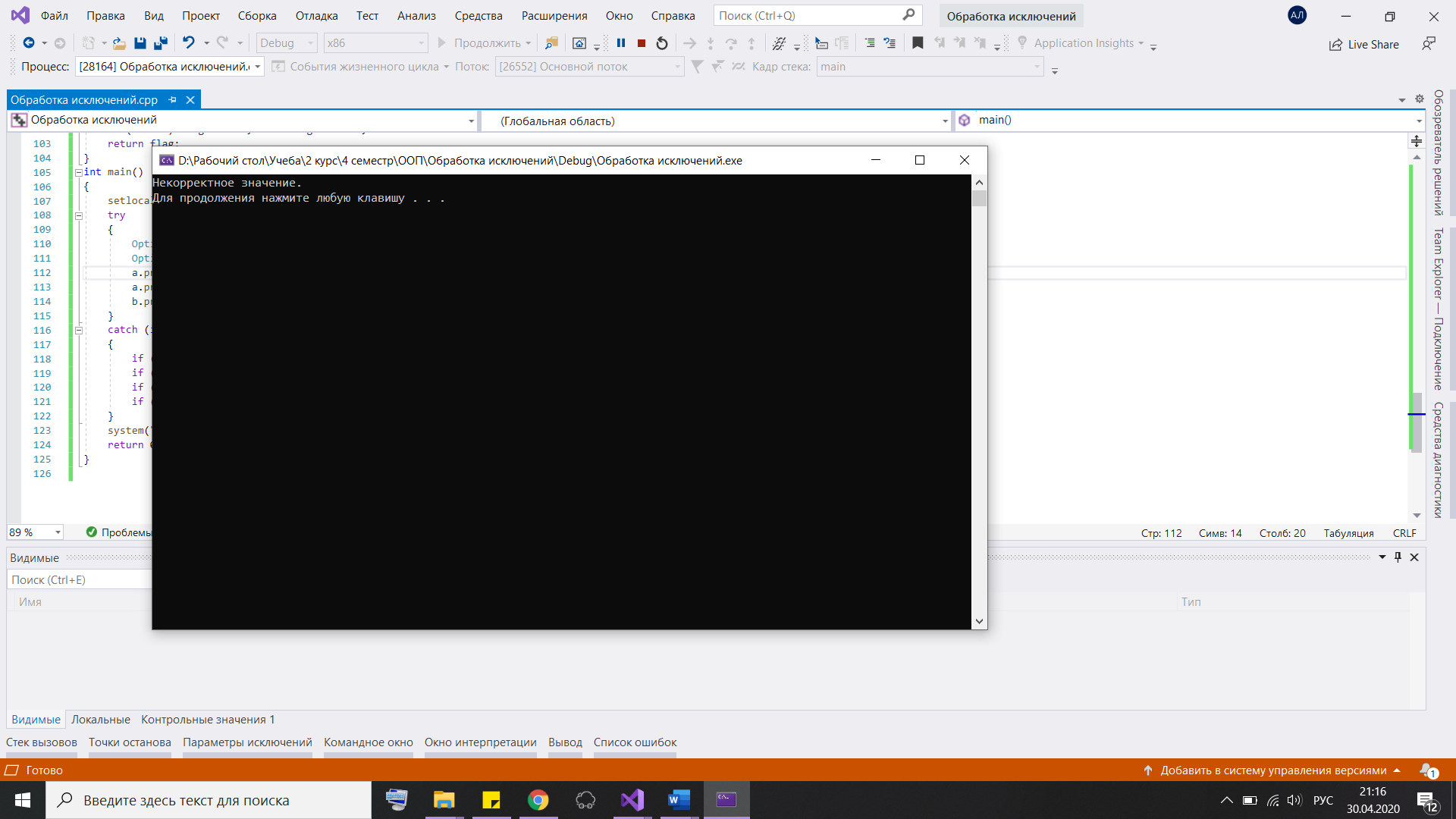
1. Optica a("Лупа", "Ближнего действия", "Выпуклая", 13, 0.2);

Optica b("Фотоаппарат", "Дальнего действия", "", 13, 0);



1. Optica a("Лупа", "Ближнего действия", "Выпуклая", 13, 0.2);

Optica b("Фотоаппарат", "Дальнего действия", "Вогнутая", 13, 0);



# Контрольные вопросы

1. Дайте определение исключения.

Обычно исключение – это возникающая в программе нештатная ситуация, с которой программа не может справиться.

1. В каком блоке описывается список обработчиков для возбужденного исключения?

Оно следует непосдерственно за try и находится после блока кода, который может генерировать исключение.

1. Опишите синтаксис обработчика исключения.

catch (формальный\_аргумент)

составная\_конструкция

1. Что такое спецификация исключения?

Синтаксически спецификация исключения является частью объявления и определения функции.

1. Для каких целей используется функция unexpected?

Предоставляемая системой функция unexpected() вызывается, когда она возбудила исключение, которое отсутствует в ее списке спецификации исключений.